Review of Proof by Principle of Mathematical Induction

Recall:

Principle of Mathematical Induction (PMI). Let P be a property of positive integers such that:

1. Base Step: P(1) is true, and
2. Inductive Step: if P(k) is true, then P(k+1) is true.

Then P(n) is true for all positive integers.

Simple example of a proof by induction:

Prove that $n^2 \in O(2^n)$.

Proof we need to find C>0 and n_0 >0 such that $n^2 \leq C * 2^n$

Let C = 1 and $n_0 = 5$ (Remember there can be many pairs, I chose this pair since $5^2 = 25$ and $2^5 = 32$ and if you think about the difference of two consecutive squares, that will be 2k+1 which will be less than $3k < k^2$ if k us bigger than 3. So 5 is a safe bet!! -- see below in the proof of the inductive case.

1. Base case: Want to show $n^2 \leq 2^n$ when n = 6
2. $6^2 = 36$ and $2^6 = 64$ since 36<64 , base case us true
3. Inductive case: need to show that $k^2 \leq 2^k \rightarrow (k+1)^2 \leq 2^{k+1}$
4. $(k+1)^2 = k^2 + 2k + 1$
   $\leq 2^k + 2k + 1$ by inductive hypothesis $\mathbf{k^2 \leq 2^k}$
   $< 2^k + k^2$ since $2k+1 < 2k+k = 3k < k^2$ if k>3
   $\leq 2^k + 2^k$ since by inductive hypothesis $\mathbf{k^2 \leq 2^k}$
5. Thus by the PMI: $n^2 \leq 2^k$ for all $n > n_0$ ($n_0 = 5$)

Example of a *stay ahead argument* for proving a greedy algorithm is correct, that is finds a compatible set with the maximum number of jobs.

Proof that the Greedy Algorithm based on choosing the job earliest finish time that is compatible with the set of chosen jobs finds a compatible set with the maximum number of jobs.

Recall the Interval Scheduling Problem: Given a set of jobs to be run on some processor specified by their start and finish times, find the largest cardinality set of jobs that are compatible. Compatible means that no two jobs run at the same time or said another way the start time of a job must be ≥ the finish time of any previous job. To make things easy we will assume that one job ending at a specified time is compatible with a job starting at the same time. Thus, the pair (job1, job2) with start and end times respectively (1:00, 2:00) and (2:00, 3:00) we will consider to be compatible.

The greedy algorithm that chooses a compatible job with earliest finish time is specified in the following pseudo code: (Note this is greedy since it is feasible, locally optimal, and irrevocable.)

```
Sort jobs by finish times so that f1 ≤ f2 ≤ ... ≤ fn.

  jobs selected
A ← ϕ
for j = 1 to n {
   if (job j compatible with A)
      A ← A ∪ {j}
}
return A
```

In many cases Greedy algorithms are proven correct using a stay ahead argument in an induction proof.
Note: There may be many optimal solutions for a given instance of the interval scheduling problem. The simplest is two overlapping intervals. A set containing either interval is optimal!

Outline of Proof: Let Σ be the set of compatible jobs given by the earliest finish time algorithm and Σ^* be some optimal solution. (There may be many optimal solutions.) Both Σ and Σ^* will be ordered by the jobs finish times. This means Σ and Σ^* are also ordered by their start times.

1. To apply Principle of Mathematical Induction we need to specify a proposition P(k) where n goes from 1 to size of the largest compatible set.
2. Assume an optimal solution exists call it $\Sigma^* = \{ (s_1^*, f_1^*), ..(s_n^*, f_n^*)\}$ with **n** compatible jobs specified by their start and end times and let $\Sigma = \{ (s_1, f_1), ..(s_m, f_m)\}$ be the set of jobs determined by the earliest finish time algorithm. Both sets are sorted by finish time.
3. By induction on P(n) we will show that our algorithm constructs a solution with n jobs.

Induction proof:

Let P(k) be the proposition $f_k \leq f^*_k$: Want to show this is true for all $k \geq 1$ up to n.
Base Case: To show P(1) is true. That is $f_1 \leq f^*_1$.
Proof of Base Case: $f_1 \leq f^*_1$ is true since f_1 has the earliest finish time of all the jobs by definition of the algorithm
Inductive case: Show that P(k) is true $\rightarrow$ P(k+1) is true for any $1 \leq k < n$. Thus we need to show that $f_k \leq f^*_k : \rightarrow f_{k+1} \leq f^*_{k+1}$:

Proof of Inductive case:

1. (s_{k+1}, f_{k+1}) is the job with the earliest finish chosen from the set of jobs that start later than (s_k, f_k)
2. Claim: job (s^*_{k+1}, f^*_{k+1}) must start after job (s_k, f_k) finishes. Since
   $s^*_{k+1} \geq f^*_k$ since (s^*_{k+1}, f^*_{k+1}) is compatible with (s^*_k, f^*_k) and
   $f^*_k \geq f_k$ by the inductive hypothesis $f_k \leq f^*_k$
   thus $s^*_{k+1} \geq f_k$
3. Thus (s^*_{k+1}, f^*_{k+1}) is compatible with (s_k, f_k)
4. Therefore (s_{k+1}, f_{k+1}) will finish no later than (s^*_{k+1}, f^*_{k+1}) or in other words $f_{k+1} \leq f^*_{k+1}$

This implies that Σ has at least as many jobs as Σ^* since the n-th job in Σ ends before the nth job in Σ^*.

Review of Proof by contradiction

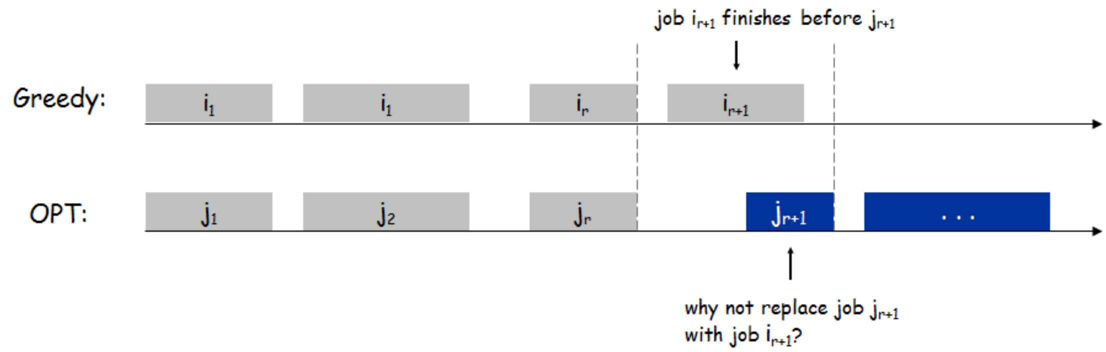
Suppose you want to prove that some theorem is true. The basic structure of a proof by contradiction is to assume that the statement is false and then contradict either one of the assumptions of the theorem you are trying to prove or some statement that you know is correct in the context of the theorem.
A simple example from high school algebra is:
Prove that the graphs of the functions defined by $y = x - 2$ and $y = x^2+2x+1$ do not intersect.
Proof:

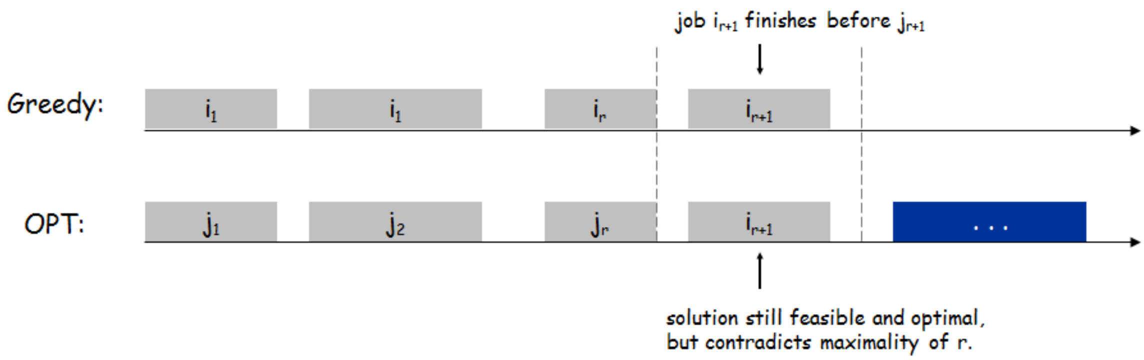
1. Suppose they two graphs do intersect.
2. Then there is some point (a, b) on both graphs.
3. Thus $b = a-2$ and $b = a^2+2a+1$
4. Thus $a-2 = a^2+2a+1$
5. Simplifying we get $0 = a^2+a+3$
6. But the discriminant ($b^2-4ac = -11$) is < 0. This implies the only solutions to this equation are not real numbers and contradicts that there is some point on both graphs,
7. Thus the graphs do not intersect

The proof by contradiction has the following structure.

1. Assume any optimal solution has **n** compatible jobs Σ
2. Let Σ be the set of jobs chosen by the earliest finish time algorithm. $|\Sigma| < n$
3. Let Σ^* be the optimal solution that has the largest number of initial jobs that agree with the jobs in Σ. That is, let $\Sigma = \{i_1, i_2, \ldots i_k\}$ and $\Sigma^* = \{j_1, j_2, \ldots j_n\}$ where $i_1 = j_1$, $i_2 = j_2$, ..., $i_r = j_r$ for the largest possible value of r.
4. If $r < \min\{n, |\Sigma|\ )$ then we will show there must be a different optimal solution, say Σ^{**} with the first r+1 jobs in common with Σ. This contradicts that Σ^* was to have the largest number of the same initial jobs.

The figure captures the situation, except it might be true that the job i_{r+1} may finish at the same time as job j_{r+1}





5. But now if $|\Sigma| < n$, there would still be jobs in Σ^* compatible with Σ, see the figure. This contradicts how the greedy earliest finish time works. It would have chosen the earliest finish time from those compatible jobs.
6. Thus $|\Sigma| = n$ and Σ has the same number of jobs as Σ^*. Thus Σ is an optimal set